BANDIT Write-up (L11 -L20)

Bandit is a character in a fictional hacking scenario on a website called OverTheWire.org.

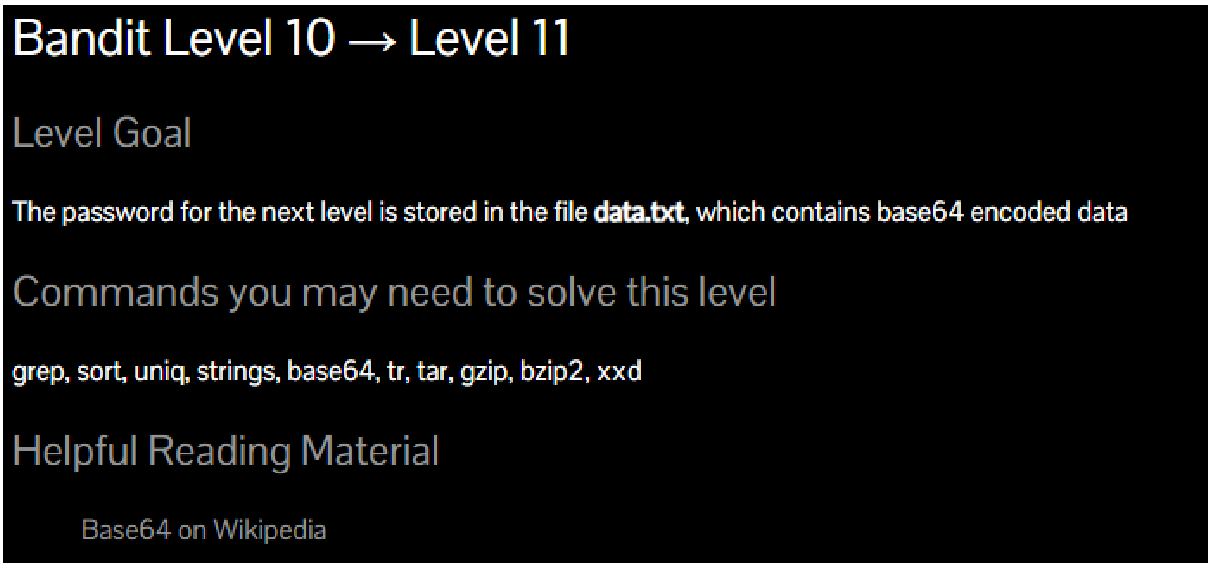
OverTheWire offers a series of wargames designed to teach cybersecurity skills in a safe environment. Bandit is the first wargame in the series, aimed at beginners. It introduces basic *Linux* commands and file manipulation through a series of challenges.

In the Bandit scenario, you play as a new user trying to gain access to higher levels by solving puzzles and cracking passwords. There is no violence or criminal activity involved.

Bandit is the suggested introductory "wargame" within the OverTheWire suite of games. It covers fundamental Linux commands and gradually progresses to advanced techniques as players advance through higher levels. Below are the walkthroughs and methodologies employed to navigate the challenges successfully.

Note for VMs: You may fail to connect to overthewire.org via SSH with a "*broken pipe error*" when the network adapter for the VM is configured to use NAT mode. Adding the setting IPQoS throughput to /etc/ssh/ssh_config should resolve the issue. If this does not solve your issue, the only option then is to change the adapter to Bridged mode.

Level 11:



Bandit Level 10 → Level 11

Level Goal

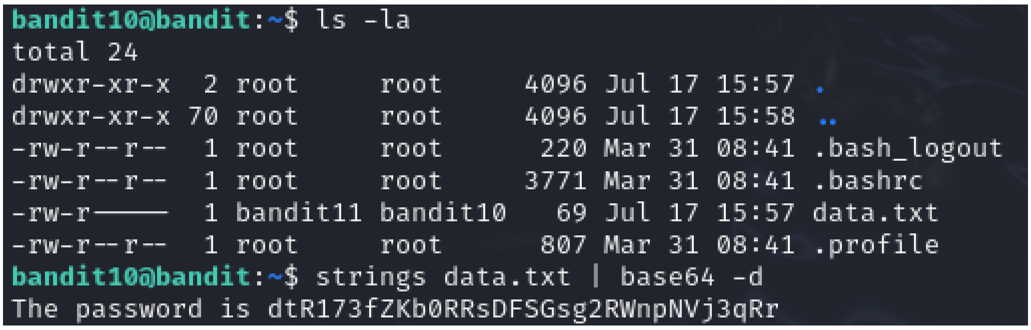
The password for the next level is stored in the file **data.txt**, which contains base64 encoded data

Commands you may need to solve this level

grep, sort, uniq, strings, base64, tr, tar, gzip, bzip2, xxd

Helpful Reading Material

Base64 on Wikipedia

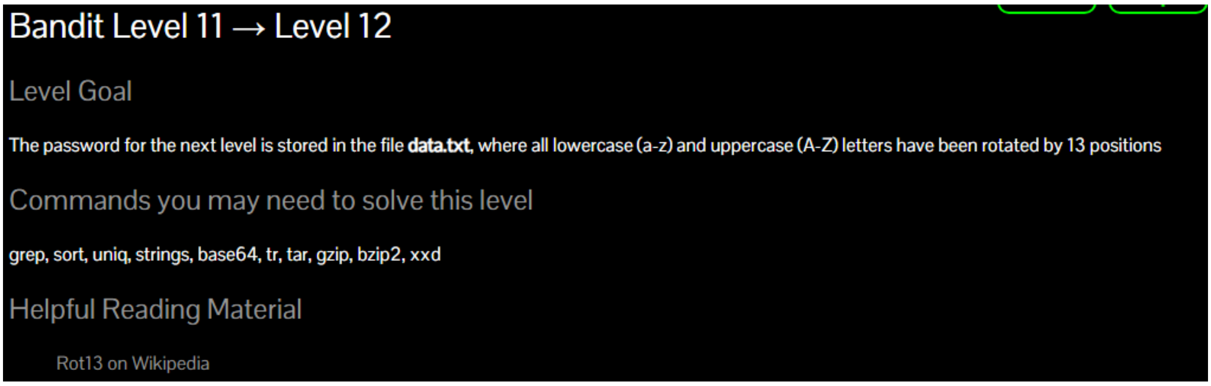


```
bandit10@bandit:~$ ls -la
total 24
drwxr-xr-x  2 root     root     4096 Jul 17 15:57 .
drwxr-xr-x 70 root     root     4096 Jul 17 15:58 ..
-rw-r--r--  1 root     root      220 Mar 31 08:41 .bash_logout
-rw-r--r--  1 root     root     3771 Mar 31 08:41 .bashrc
-rw-r-----  1 bandit11 bandit10   69 Jul 17 15:57 data.txt
-rw-r--r--  1 root     root      807 Mar 31 08:41 .profile
bandit10@bandit:~$ strings data.txt | base64 -d
The password is dtR173fZKb0RRsDFSGsg2RWnpNVj3qRr
```

We will print only printable strings in the data.txt and base64 -d will decode the base64 string.

`dtR173fZKb0RRsDFSGsg2RWnpNVj3qRr`

Level 12:



Bandit Level 11 → Level 12

Level Goal

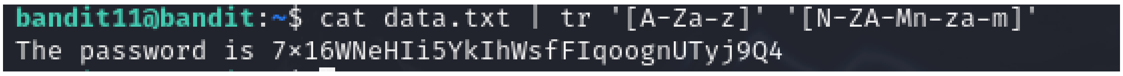
The password for the next level is stored in the file **data.txt**, where all lowercase (a-z) and uppercase (A-Z) letters have been rotated by 13 positions

Commands you may need to solve this level

grep, sort, uniq, strings, base64, tr, tar, gzip, bzip2, xxd

Helpful Reading Material

- Rot13 on Wikipedia



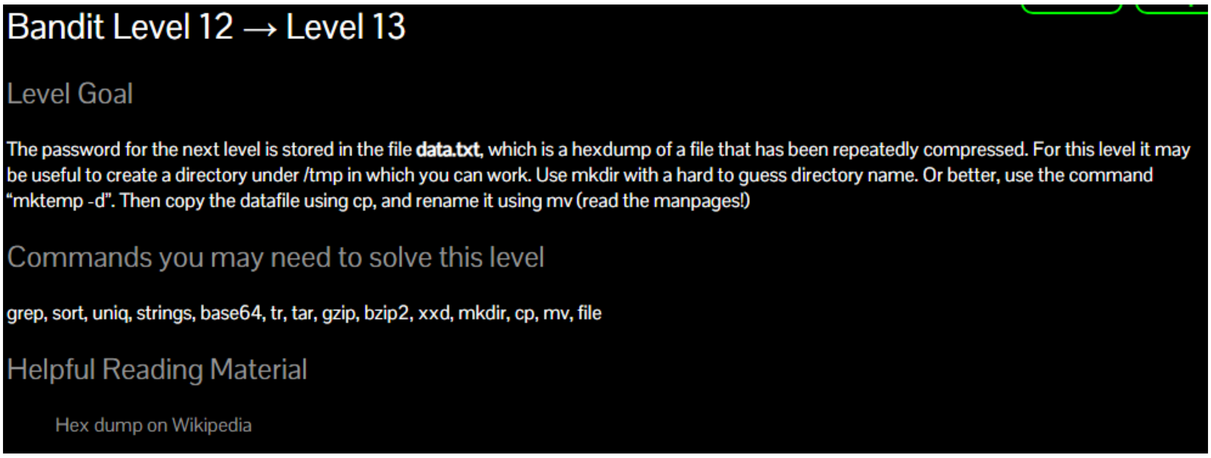
```
bandit11@bandit:~$ cat data.txt | tr '[A-Za-z]' '[N-ZA-Mn-za-m]'
The password is 7x16WNeHIi5YkIhWsfFIqoognUTyj9Q4
```

This command will print data.txt but rotate each letter (capital and small) to the 13th position.

`7x16WNeHIi5YkIhWsfFIqoognUTyj9Q4`

And see 7k16JArUVv5LxVuJfsSVdbbtaHGlw9D4 is rotated 13th position next above.

Level 13:



Bandit Level 12 → Level 13

Level Goal

The password for the next level is stored in the file **data.txt**, which is a hexdump of a file that has been repeatedly compressed. For this level it may be useful to create a directory under /tmp in which you can work. Use mkdir with a hard to guess directory name. Or better, use the command "mktemp -d". Then copy the datafile using cp, and rename it using mv (read the manpages!)

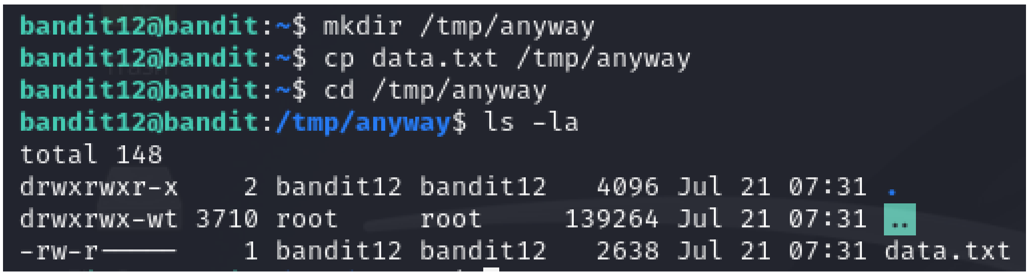
Commands you may need to solve this level

grep, sort, uniq, strings, base64, tr, tar, gzip, bzip2, xxd, mkdir, cp, mv, file

Helpful Reading Material

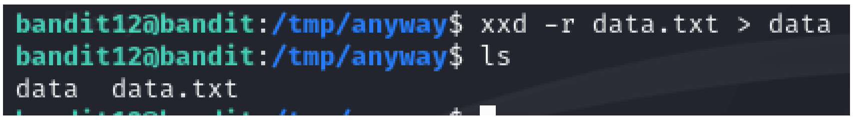
Hex dump on Wikipedia

For this task, it is recommended to perform this lab in another directory:



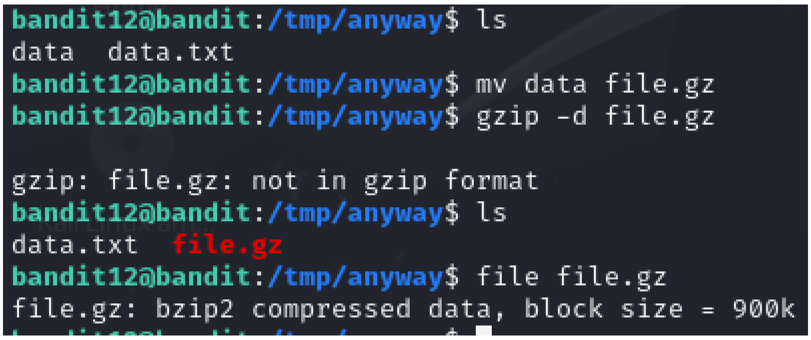
```
bandit12@bandit:~$ mkdir /tmp/anyway
bandit12@bandit:~$ cp data.txt /tmp/anyway
bandit12@bandit:~$ cd /tmp/anyway
bandit12@bandit:/tmp/anyway$ ls -la
total 148
drwxrwxr-x    2 bandit12 bandit12   4096 Jul 21 07:31 .
drwxrwx-wt 3710 root     root     139264 Jul 21 07:31 ..
-rw-r-----    1 bandit12 bandit12   2638 Jul 21 07:31 data.txt
```

We use xxd, which will either hashdump or reverse the text, with -r it will reverse. And data.txt to data:

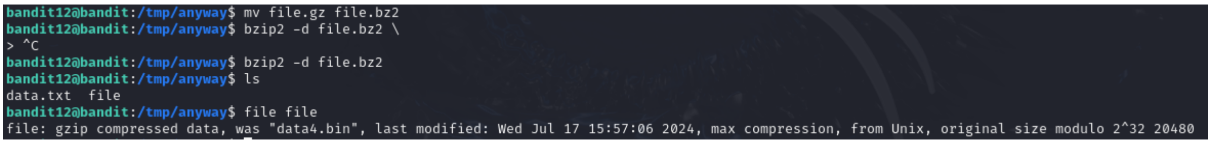


```
bandit12@bandit:/tmp/anyway$ xxd -r data.txt > data
bandit12@bandit:/tmp/anyway$ ls
data  data.txt
```

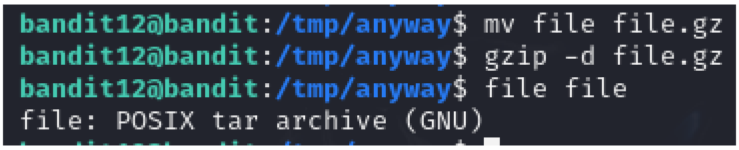
As this file is repeatedly compressed, we need to repeatedly decompress it until it completes, we'll do this by gzip and bzip2 where required, each time by changing the extension of the file that we have revered before:



```
bandit12@bandit:/tmp/anyway$ ls
data  data.txt
bandit12@bandit:/tmp/anyway$ mv data file.gz
bandit12@bandit:/tmp/anyway$ gzip -d file.gz

gzip: file.gz: not in gzip format
bandit12@bandit:/tmp/anyway$ ls
data.txt  file.gz
bandit12@bandit:/tmp/anyway$ file file.gz
file.gz: bzip2 compressed data, block size = 900k
```

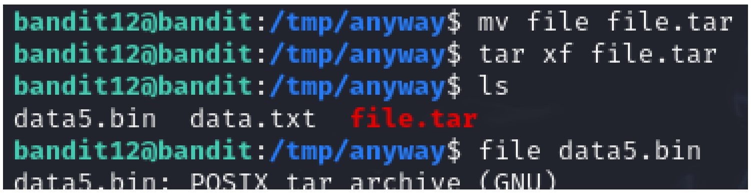


```
bandit12@bandit:/tmp/anyway$ mv file.gz file.bz2
bandit12@bandit:/tmp/anyway$ bzip2 -d file.bz2 \
> ^C
bandit12@bandit:/tmp/anyway$ bzip2 -d file.bz2
bandit12@bandit:/tmp/anyway$ ls
data.txt  file
bandit12@bandit:/tmp/anyway$ file file
file: gzip compressed data, was "data4.bin", last modified: Wed Jul 17 15:57:06 2024, max compression, from Unix, original size modulo 2^32 20480
```



```
bandit12@bandit:/tmp/anyway$ mv file file.gz
bandit12@bandit:/tmp/anyway$ gzip -d file.gz
bandit12@bandit:/tmp/anyway$ file file
file: POSIX tar archive (GNU)
```

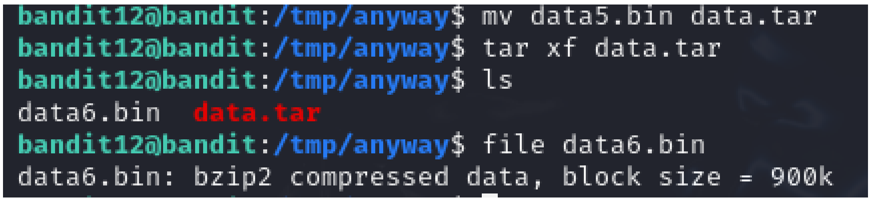
Now we'll use an extracting tool - tar:



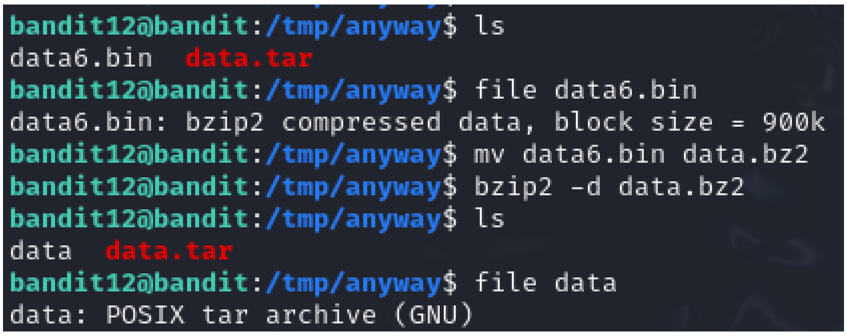
```
bandit12@bandit:/tmp/anyway$ mv file file.tar
bandit12@bandit:/tmp/anyway$ tar xf file.tar
bandit12@bandit:/tmp/anyway$ ls
data5.bin  data.txt  file.tar
bandit12@bandit:/tmp/anyway$ file data5.bin
data5.bin: POSIX tar archive (GNU)
```

We can delete both files other than ***data5.bin***.

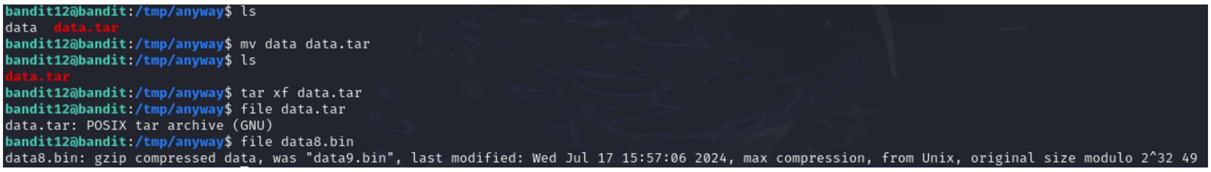
Now we need to again repeatedly decompress this file to get an ASCII, instead of repeated gz, bz and tar format



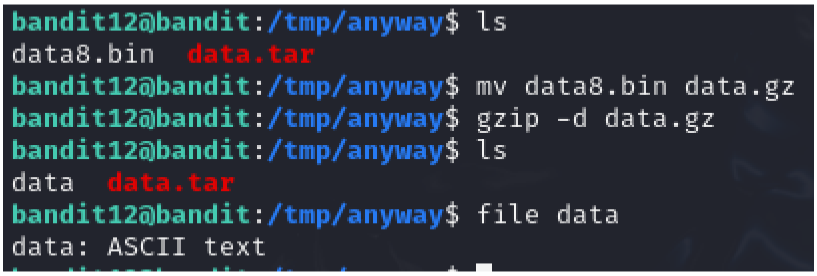
```
bandit12@bandit:/tmp/anyway$ mv data5.bin data.tar
bandit12@bandit:/tmp/anyway$ tar xf data.tar
bandit12@bandit:/tmp/anyway$ ls
data6.bin  data.tar
bandit12@bandit:/tmp/anyway$ file data6.bin
data6.bin: bzip2 compressed data, block size = 900k
```



```
bandit12@bandit:/tmp/anyway$ ls
data6.bin  data.tar
bandit12@bandit:/tmp/anyway$ file data6.bin
data6.bin: bzip2 compressed data, block size = 900k
bandit12@bandit:/tmp/anyway$ mv data6.bin data.bz2
bandit12@bandit:/tmp/anyway$ bzip2 -d data.bz2
bandit12@bandit:/tmp/anyway$ ls
data  data.tar
bandit12@bandit:/tmp/anyway$ file data
data: POSIX tar archive (GNU)
```



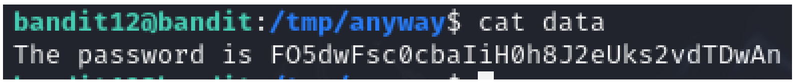
```
bandit12@bandit:/tmp/anyway$ ls
data  data.tar
bandit12@bandit:/tmp/anyway$ mv data data.tar
bandit12@bandit:/tmp/anyway$ ls
data.tar
bandit12@bandit:/tmp/anyway$ tar xf data.tar
bandit12@bandit:/tmp/anyway$ file data.tar
data.tar: POSIX tar archive (GNU)
bandit12@bandit:/tmp/anyway$ file data8.bin
data8.bin: gzip compressed data, was "data9.bin", last modified: Wed Jul 17 15:57:06 2024, max compression, from Unix, original size modulo 2^32 49
```



```
bandit12@bandit:/tmp/anyway$ ls
data8.bin  data.tar
bandit12@bandit:/tmp/anyway$ mv data8.bin data.gz
bandit12@bandit:/tmp/anyway$ gzip -d data.gz
bandit12@bandit:/tmp/anyway$ ls
data  data.tar
bandit12@bandit:/tmp/anyway$ file data
data: ASCII text
```

Finally, by unpacking, again and again, we have got an ASCII format.

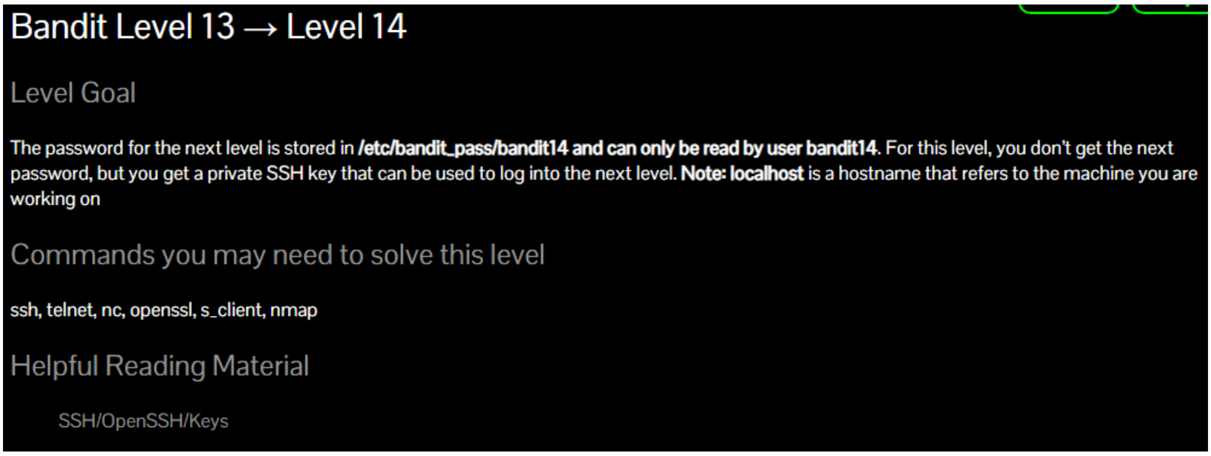
Just cat this file:



```
bandit12@bandit:/tmp/anyway$ cat data
The password is FO5dwFsc0cbaIiH0h8J2eUks2vdTDwAn
bandit12@bandit:/tmp/anyway$
```

`FO5dwFsc0cbaIiH0h8J2eUks2vdTDwAn`

Level 14:



Bandit Level 13 → Level 14

Level Goal

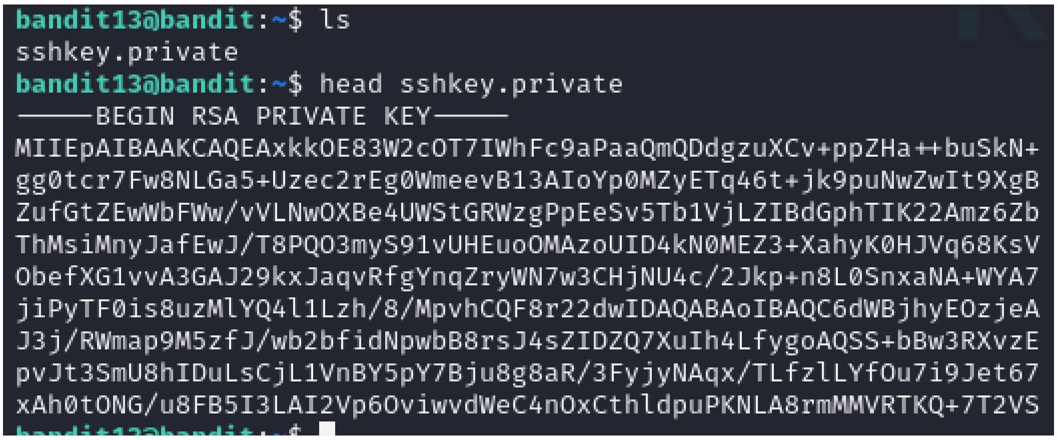
The password for the next level is stored in **/etc/bandit_pass/bandit14 and can only be read by user bandit14**. For this level, you don't get the next password, but you get a private SSH key that can be used to log into the next level. **Note: localhost** is a hostname that refers to the machine you are working on

Commands you may need to solve this level

ssh, telnet, nc, openssl, s_client, nmap

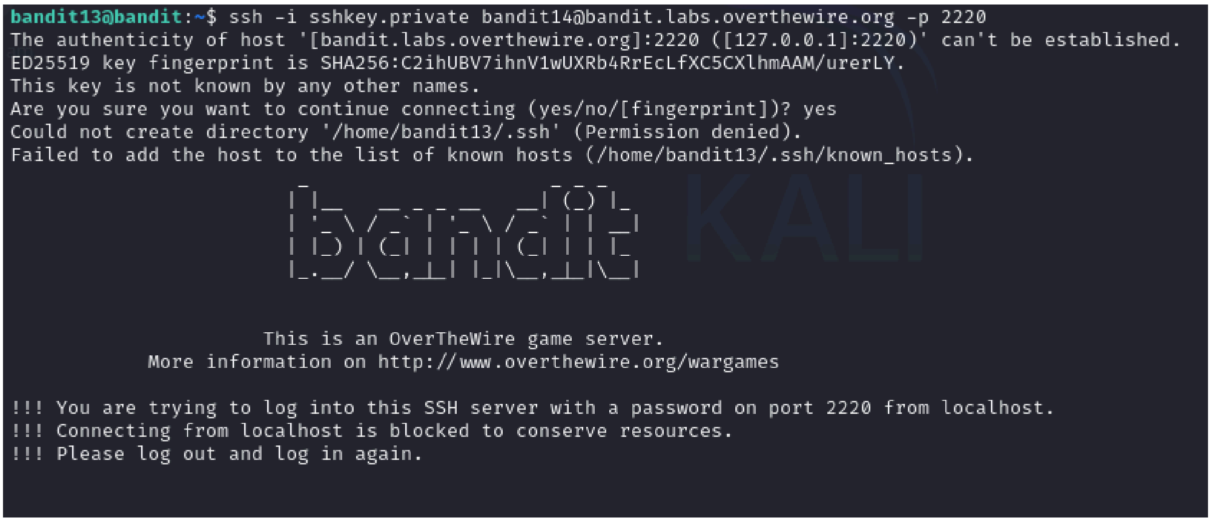
Helpful Reading Material

SSH/OpenSSH/Keys



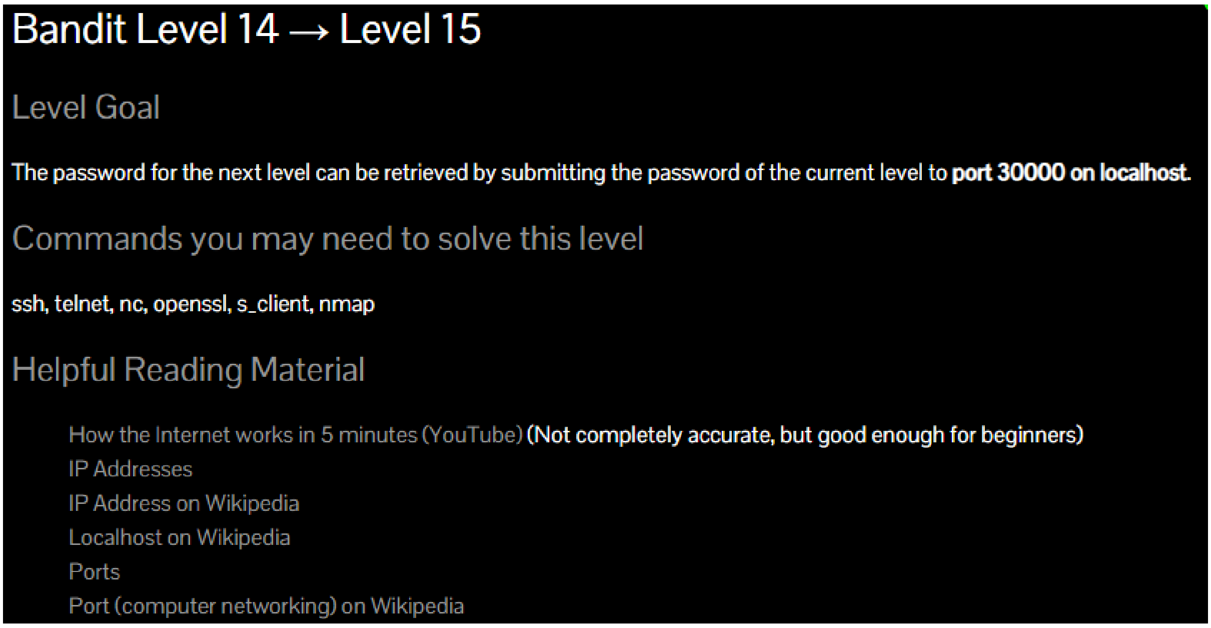
```
bandit13@bandit:~$ ls
sshkey.private
bandit13@bandit:~$ head sshkey.private
-----BEGIN RSA PRIVATE KEY-----
MIIEpAIBAAKCAQEAxkkOE83W2cOT7IWhFc9aPaaQmQDdgzuXCv+ppZHa++buSkN+
gg0tcr7Fw8NLGa5+Uzec2rEg0WmeevB13AIoYp0MZyETq46t+jk9puNwZwIt9XgB
ZufGtZEwWbFWw/vVLNwOXBe4UWStGRWzgPpEeSv5Tb1VjLZIBdGphTIK22Amz6Zb
ThMsiMnyJafEwJ/T8PQO3myS91vUHEuoOMAzoUID4kN0MEZ3+XahyK0HJVq68KsV
ObefXG1vvA3GAJ29kxJaqvRfgYnqZryWN7w3CHjNU4c/2Jkp+n8L0SnxaNA+WYA7
jiPyTF0is8uzMlYQ4l1Lzh/8/MpvhCQF8r22dwIDAQABAoIBAQC6dWBjhyEOzjeA
J3j/RWmap9M5zfJ/wb2bfidNpwbB8rsJ4sZIDZQ7XuIh4LfygoAQSS+bBw3RXvzE
pvJt3SmU8hIDuLsCjL1VnBY5pY7Bju8g8aR/3FyjyNAqx/TLfzlLYfOu7i9Jet67
xAh0tONG/u8FB5I3LAI2Vp6OviwvdWeC4nOxCthldpuPKNLA8rmMMVRTKQ+7T2VS
bandit13@bandit:~$
```

So it only has an SSH key, which we can use to gain access to the next level:



```
bandit13@bandit:~$ ssh -i sshkey.private bandit14@bandit.labs.overthewire.org -p 2220
The authenticity of host '[bandit.labs.overthewire.org]:2220 ([127.0.0.1]:2220)' can't be established.
ED25519 key fingerprint is SHA256:C2ihUBV7ihnV1wUXRb4RrEcLfXC5CXlhmAAM/urerLY.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Could not create directory '/home/bandit13/.ssh' (Permission denied).
Failed to add the host to the list of known hosts (/home/bandit13/.ssh/known_hosts).
                         _                     _ _ _
                        | |__   __ _ _ __   __| (_) |_
                        | '_ \ / _` | '_ \ / _` | | __|
                        | |_) | (_| | | | | (_| | | |_
                        |_.__/ \__,_|_| |_|\__,_|_|\__|


                      This is an OverTheWire game server.
            More information on http://www.overthewire.org/wargames

!!! You are trying to log into this SSH server with a password on port 2220 from localhost.
!!! Connecting from localhost is blocked to conserve resources.
!!! Please log out and log in again.
```

Level 15:



Bandit Level 14 → Level 15

Level Goal

The password for the next level can be retrieved by submitting the password of the current level to **port 30000 on localhost.**

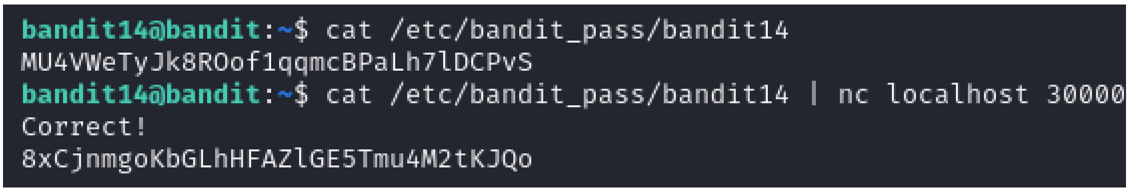
Commands you may need to solve this level

ssh, telnet, nc, openssl, s_client, nmap

Helpful Reading Material

- How the Internet works in 5 minutes (YouTube) **(Not completely accurate, but good enough for beginners)**
- IP Addresses
- IP Address on Wikipedia
- Localhost on Wikipedia
- Ports
- Port (computer networking) on Wikipedia

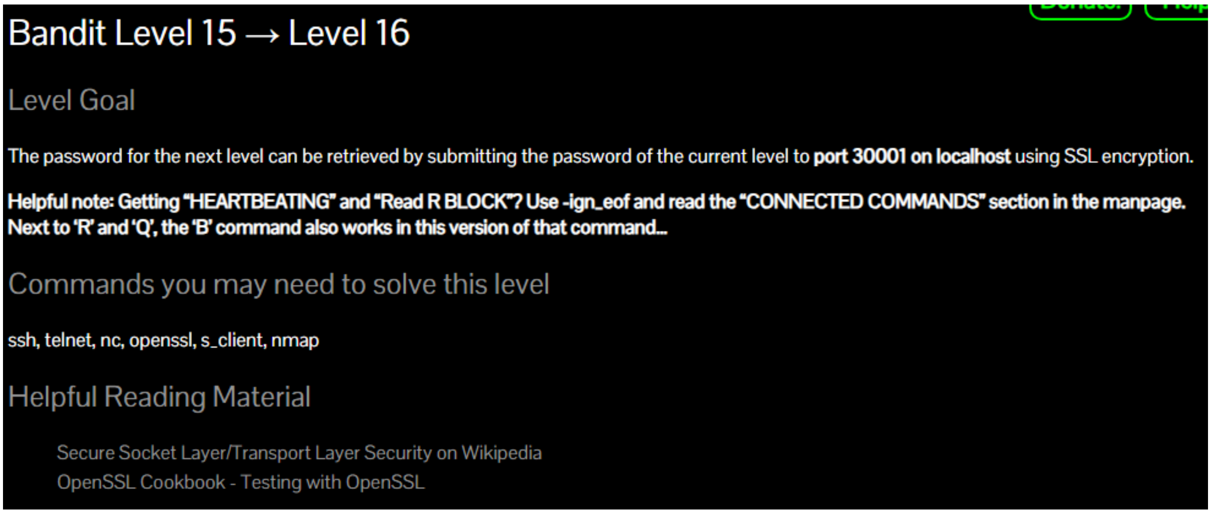
So password for the next level can be found by retrieving the password of the current level on port 300000 local host. We know that the current password is located at: */etc/bandit_pass/bandit14*.



```
bandit14@bandit:~$ cat /etc/bandit_pass/bandit14
MU4VWeTyJk8ROof1qqmcBPaLh7lDCPvS
bandit14@bandit:~$ cat /etc/bandit_pass/bandit14 | nc localhost 30000
Correct!
8xCjnmgoKbGLhHFAZlGE5Tmu4M2tKJQo
```

8xCjnmgoKbGLhHFAZlGE5Tmu4M2tKJQo

Level 16:



Bandit Level 15 → Level 16

Level Goal

The password for the next level can be retrieved by submitting the password of the current level to **port 30001 on localhost** using SSL encryption.

Helpful note: Getting "HEARTBEATING" and "Read R BLOCK"? Use -ign_eof and read the "CONNECTED COMMANDS" section in the manpage. Next to 'R' and 'Q', the 'B' command also works in this version of that command…

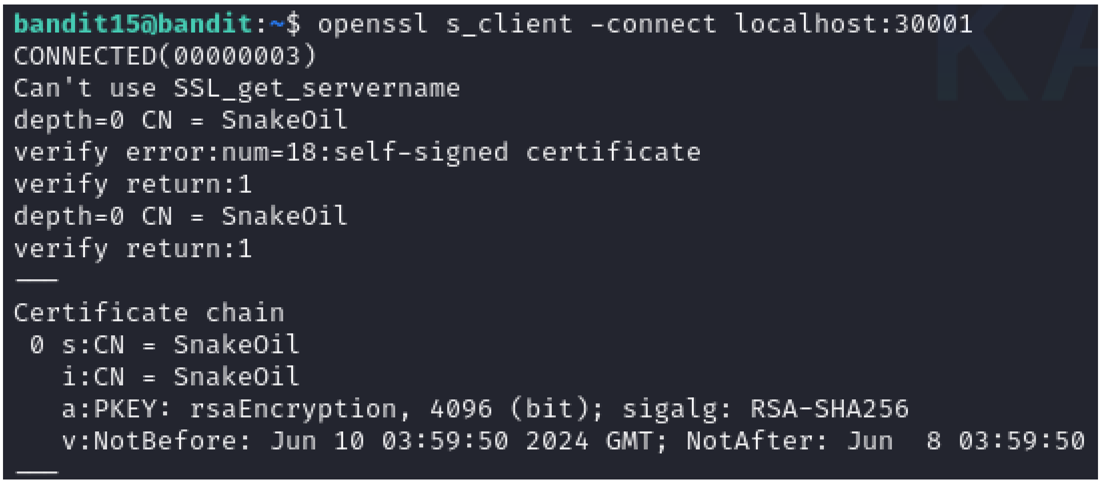
Commands you may need to solve this level

ssh, telnet, nc, openssl, s_client, nmap

Helpful Reading Material

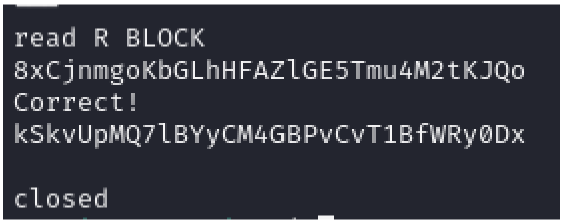
- Secure Socket Layer/Transport Layer Security on Wikipedia
- OpenSSL Cookbook - Testing with OpenSSL

Password for the next level can be retrieved if we submit the current level's password to port 300001 localhost using OpenSSL:



```
bandit15@bandit:~$ openssl s_client -connect localhost:30001
CONNECTED(00000003)
Can't use SSL_get_servername
depth=0 CN = SnakeOil
verify error:num=18:self-signed certificate
verify return:1
depth=0 CN = SnakeOil
verify return:1
---
Certificate chain
 0 s:CN = SnakeOil
   i:CN = SnakeOil
   a:PKEY: rsaEncryption, 4096 (bit); sigalg: RSA-SHA256
   v:NotBefore: Jun 10 03:59:50 2024 GMT; NotAfter: Jun  8 03:59:50
---
```

s_client: This subcommand of OpenSSL is used to establish a TLS/SSL client connection.

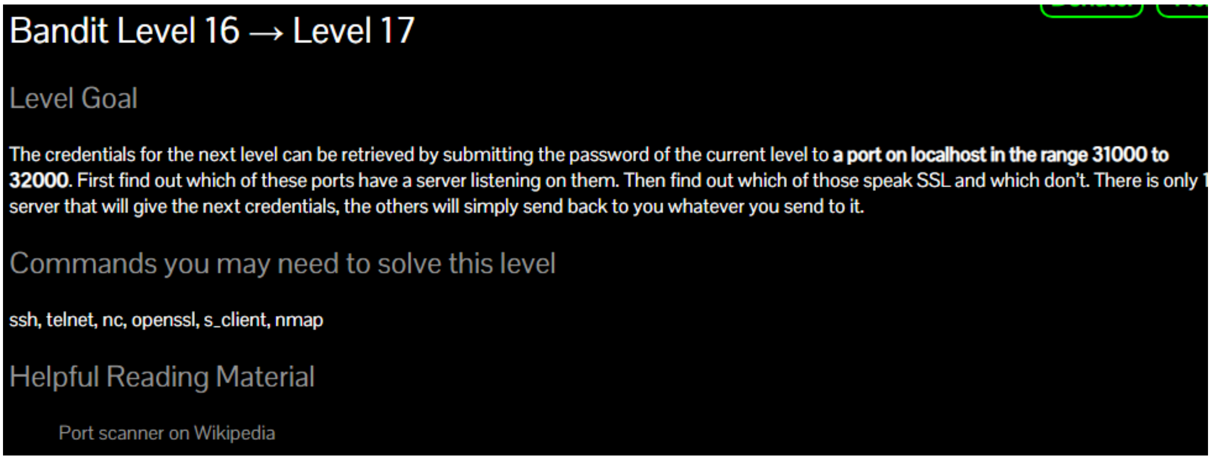


```
read R BLOCK
8xCjnmgoKbGLhHFAZlGE5Tmu4M2tKJQo
Correct!
kSkvUpMQ7lBYyCM4GBPvCvT1BfWRy0Dx

closed
```

If we enter the current password here, it will give us a flag.

kSkvUpMQ7lBYyCM4GBPvCvT1BfWRy0Dx

Level 17:



Bandit Level 16 → Level 17

Level Goal

The credentials for the next level can be retrieved by submitting the password of the current level to **a port on localhost in the range 31000 to 32000**. First find out which of these ports have a server listening on them. Then find out which of those speak SSL and which don't. There is only 1 server that will give the next credentials, the others will simply send back to you whatever you send to it.

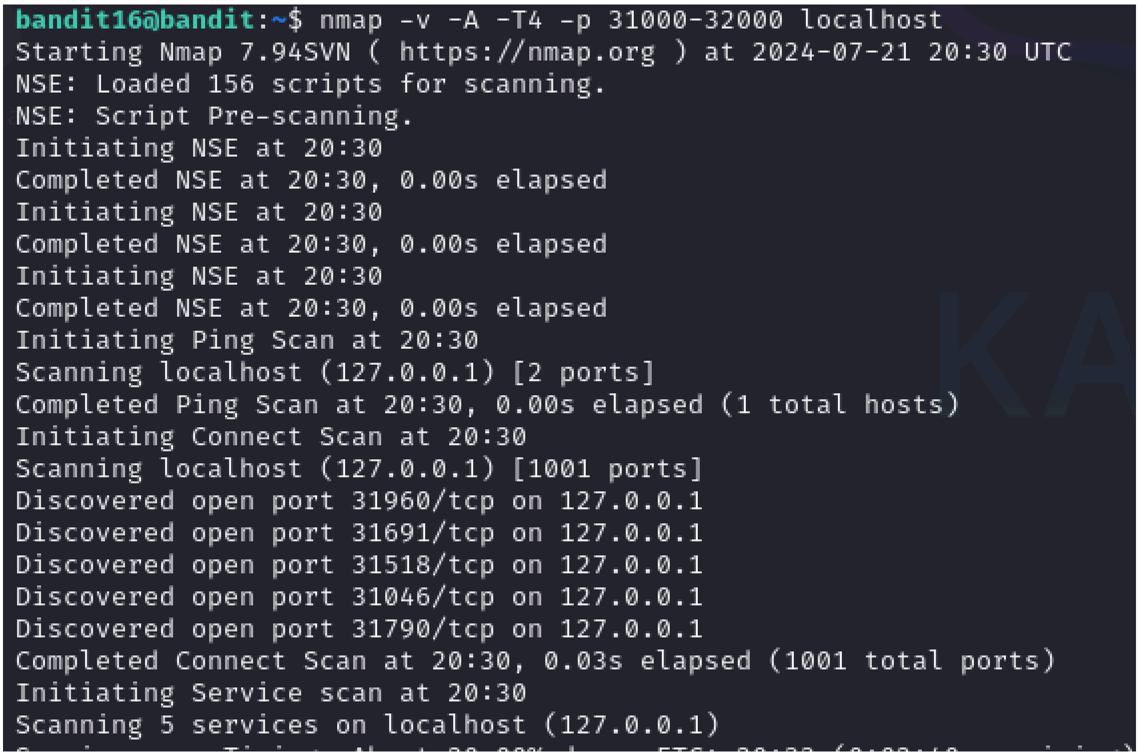
Commands you may need to solve this level

ssh, telnet, nc, openssl, s_client, nmap

Helpful Reading Material

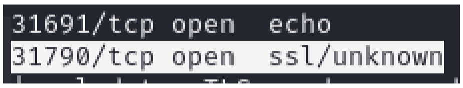
Port scanner on Wikipedia

So here we have several, we need to see which one is active and which is running SSL service. We use the same command as the previous level for that port to get the flag.



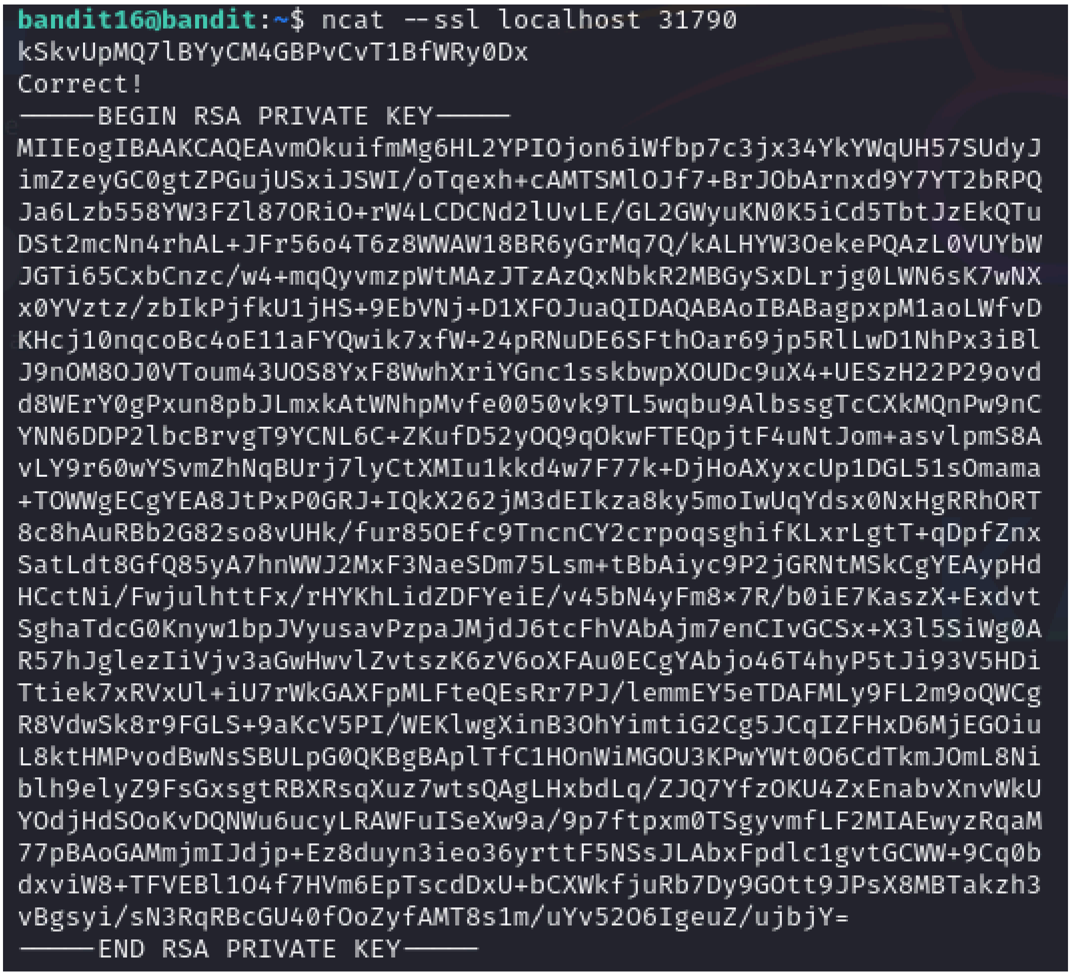
```
bandit16@bandit:~$ nmap -v -A -T4 -p 31000-32000 localhost
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-07-21 20:30 UTC
NSE: Loaded 156 scripts for scanning.
NSE: Script Pre-scanning.
Initiating NSE at 20:30
Completed NSE at 20:30, 0.00s elapsed
Initiating NSE at 20:30
Completed NSE at 20:30, 0.00s elapsed
Initiating NSE at 20:30
Completed NSE at 20:30, 0.00s elapsed
Initiating Ping Scan at 20:30
Scanning localhost (127.0.0.1) [2 ports]
Completed Ping Scan at 20:30, 0.00s elapsed (1 total hosts)
Initiating Connect Scan at 20:30
Scanning localhost (127.0.0.1) [1001 ports]
Discovered open port 31960/tcp on 127.0.0.1
Discovered open port 31691/tcp on 127.0.0.1
Discovered open port 31518/tcp on 127.0.0.1
Discovered open port 31046/tcp on 127.0.0.1
Discovered open port 31790/tcp on 127.0.0.1
Completed Connect Scan at 20:30, 0.03s elapsed (1001 total ports)
Initiating Service scan at 20:30
Scanning 5 services on localhost (127.0.0.1)
```

We used Nmap for this step, with verbose (v), aggressive scanning (A), timing template 4 (T4), and port range as specified.



```
31691/tcp open  echo
31790/tcp open  ssl/unknown
```

On port 31790 there is SSL running.

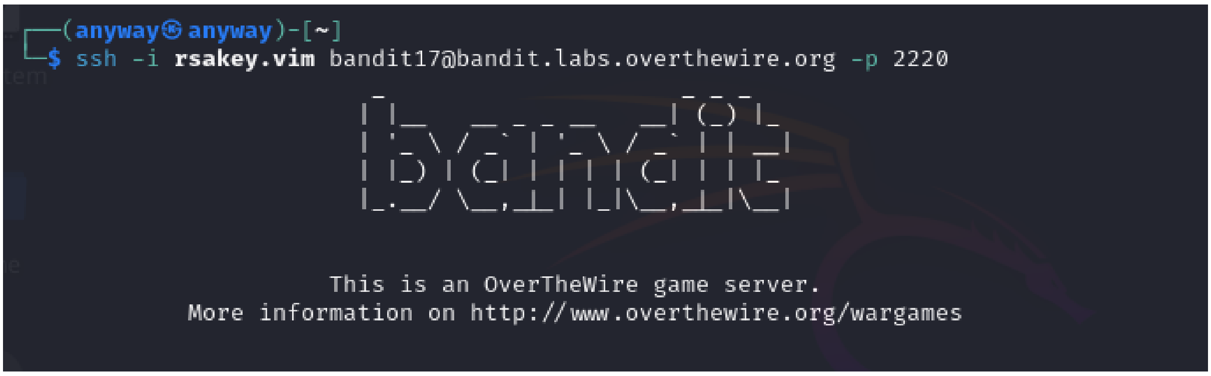


```
bandit16@bandit:~$ ncat --ssl localhost 31790
kSkvUpMQ7lBYyCM4GBPvCvT1BfWRy0Dx
Correct!
-----BEGIN RSA PRIVATE KEY-----
MIIEogIBAAKCAQEAvmOkuifmMg6HL2YPIOjon6iWfbp7c3jx34YkYWqUH57SUdyJ
imZzeyGC0gtZPGujUSxiJSWI/oTqexh+cAMTSMlOJf7+BrJObArnxd9Y7YT2bRPQ
Ja6Lzb558YW3FZl87ORiO+rW4LCDCNd2lUvLE/GL2GWyuKN0K5iCd5TbtJzEkQTu
DSt2mcNn4rhAL+JFr56o4T6z8WWAW18BR6yGrMq7Q/kALHYW3OekePQAzL0VUYbW
JGTi65CxbCnzc/w4+mqQyvmzpWtMAzJTzAzQxNbkR2MBGySxDLrjg0LWN6sK7wNX
x0YVztz/zbIkPjfkU1jHS+9EbVNj+D1XFOJuaQIDAQABAoIBABagpxpM1aoLWfvD
KHcj10nqcoBc4oE11aFYQwik7xfW+24pRNuDE6SFthOar69jp5RlLwD1NhPx3iBl
J9nOM8OJ0VToum43UOS8YxF8WwhXriYGnc1sskbwpXOUDc9uX4+UESzH22P29ovd
d8WErY0gPxun8pbJLmxkAtWNhpMvfe0050vk9TL5wqbu9AlbssgTcCXkMQnPw9nC
YNN6DDP2lbcBrvgT9YCNL6C+ZKufD52yOQ9qOkwFTEQpjtF4uNtJom+asvlpmS8A
vLY9r60wYSvmZhNqBUrj7lyCtXMIu1kkd4w7F77k+DjHoAXyxcUp1DGL51sOmama
+TOWWgECgYEA8JtPxP0GRJ+IQkX262jM3dEIkza8ky5moIwUqYdsx0NxHgRRhORT
8c8hAuRBb2G82so8vUHk/fur85OEfc9TncnCY2crpoqsghifKLxrLgtT+qDpfZnx
SatLdt8GfQ85yA7hnWWJ2MxF3NaeSDm75Lsm+tBbAiyc9P2jGRNtMSkCgYEAypHd
HCctNi/FwjulhttFx/rHYKhLidZDFYeiE/v45bN4yFm8x7R/b0iE7KaszX+Exdvt
SghaTdcG0Knyw1bpJVyusavPzpaJMjdJ6tcFhVAbAjm7enCIvGCSx+X3l5SiWg0A
R57hJglezIiVjv3aGwHwvlZvtszK6zV6oXFAu0ECgYAbjo46T4hyP5tJi93V5HDi
Ttiek7xRVxUl+iU7rWkGAXFpMLFteQEsRr7PJ/lemmEY5eTDAFMLy9FL2m9oQWCg
R8VdwSk8r9FGLS+9aKcV5PI/WEKlwgXinB3OhYimtiG2Cg5JCqIZFHxD6MjEGOiu
L8ktHMPvodBwNsSBULpG0QKBgBAplTfC1HOnWiMGOU3KPwYWt0O6CdTkmJOmL8Ni
blh9elyZ9FsGxsgtRBXRsqXuz7wtsQAgLHxbdLq/ZJQ7YfzOKU4ZxEnabvXnvWkU
YOdjHdSOoKvDQNWu6ucyLRAWFuISeXw9a/9p7ftpxm0TSgyvmfLF2MIAEwyzRqaM
77pBAoGAMmjmIJdjp+Ez8duyn3ieo36yrttF5NSsJLAbxFpdlc1gvtGCWW+9Cq0b
dxviW8+TFVEBl1O4f7HVm6EpTscdDxU+bCXWkfjuRb7Dy9GOtt9JPsX8MBTakzh3
vBgsyi/sN3RqRBcGU40fOoZyfAMT8s1m/uYv52O6IgeuZ/ujbjY=
-----END RSA PRIVATE KEY-----
```

This command is an alternative to the command we used in the previous level. And it is a simple one.

We will log in to Bandit 17 using this Private RSA key. First, save this key to a text file or vim file. Then give it to read by owner only permission by using: ***chmod 400 rsakey.vim***

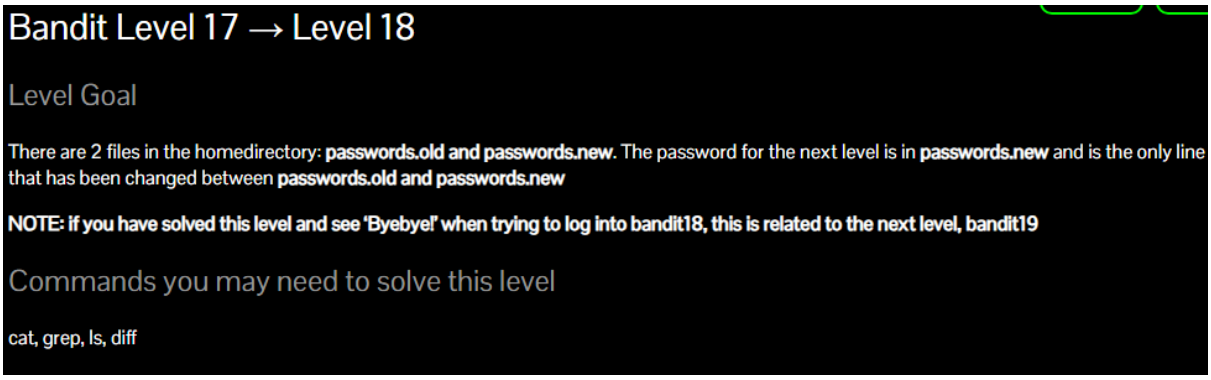
Now login to Bandit 17 using this RSA key:



```
┌──(anyway㉿anyway)-[~]
└─$ ssh -i rsakey.vim bandit17@bandit.labs.overthewire.org -p 2220
                         _                     _ _ _
                        | |__   __ _ _ __   __| (_) |_
                        | '_ \ / _` | '_ \ / _` | | __|
                        | |_) | (_| | | | | (_| | | |_
                        |_.__/ \__,_|_| |_|\__,_|_|\__|


                      This is an OverTheWire game server.
            More information on http://www.overthewire.org/wargames
```

It has logged in without asking for a password.

Level 18:



Bandit Level 17 → Level 18

Level Goal

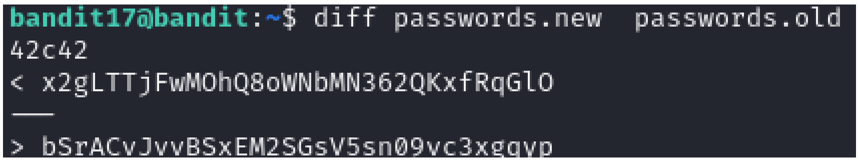
There are 2 files in the homedirectory: **passwords.old and passwords.new**. The password for the next level is in **passwords.new** and is the only line that has been changed between **passwords.old and passwords.new**

NOTE: if you have solved this level and see 'Byebye!' when trying to log into bandit18, this is related to the next level, bandit19

Commands you may need to solve this level

cat, grep, ls, diff

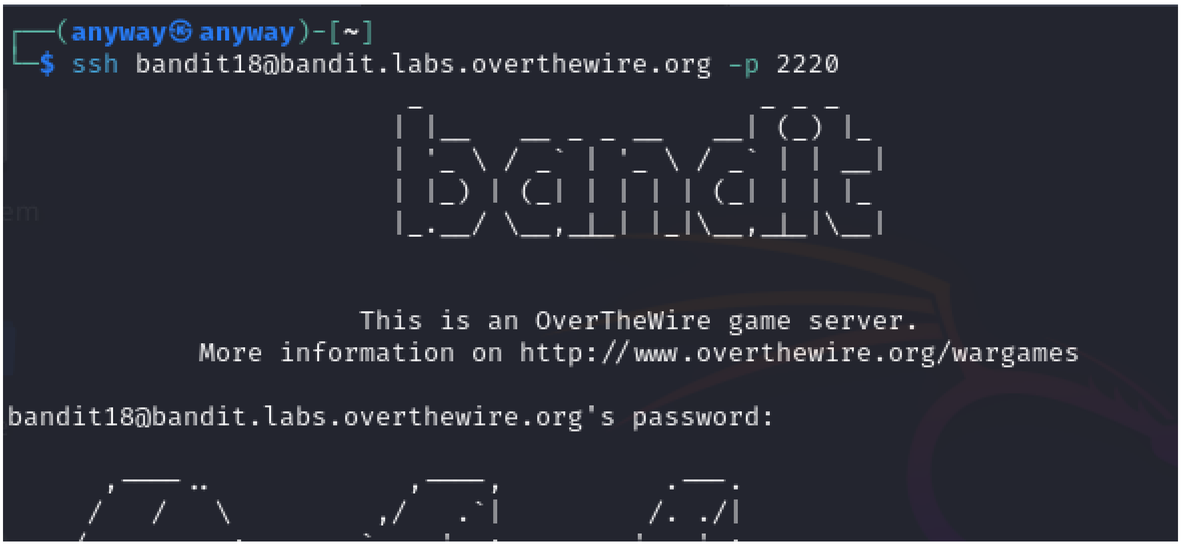
So in this level, there are two files with one difference of line in the password.new which is a password for bandit19:



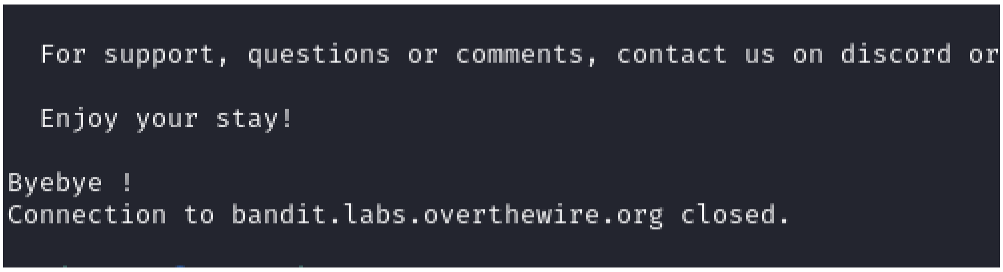
```
bandit17@bandit:~$ diff passwords.new  passwords.old
42c42
< x2gLTTjFwMOhQ8oWNbMN362QKxfRqGlO
---
> bSrACvJvvBSxEM2SGsV5sn09vc3xgqvp
```

We used diff command to see different lines in those two files.

x2gLTTjFwMOhQ8oWNbMN362QKxfRqGlO - This is the line that is changed and it's our flag.



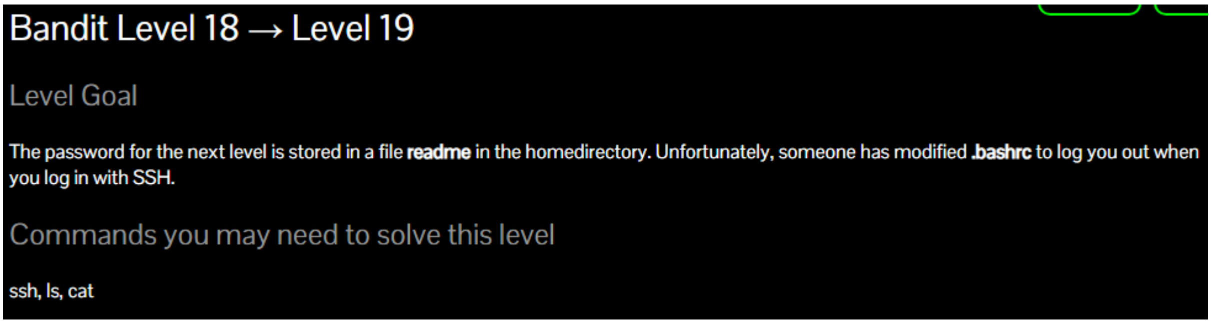
```
┌──(anyway㉿anyway)-[~]
└─$ ssh bandit18@bandit.labs.overthewire.org -p 2220
                         _                     _ _ _
                        | |__   __ _ _ __   __| (_) |_
                        | '_ \ / _` | '_ \ / _` | | __|
                        | |_) | (_| | | | | (_| | | |_
                        |_.__/ \__,_|_| |_|\__,_|_|\__|


                      This is an OverTheWire game server.
            More information on http://www.overthewire.org/wargames

bandit18@bandit.labs.overthewire.org's password:

      ,----..            ,----,          .---.
     /   /   \         ,/   .`|         /. ./|
```



```
  For support, questions or comments, contact us on discord or

  Enjoy your stay!

Byebye !
Connection to bandit.labs.overthewire.org closed.
```

Previously mentioned that if bandit18 shows BYEBYE we should skip to the next level.

Level 19:



Bandit Level 18 → Level 19

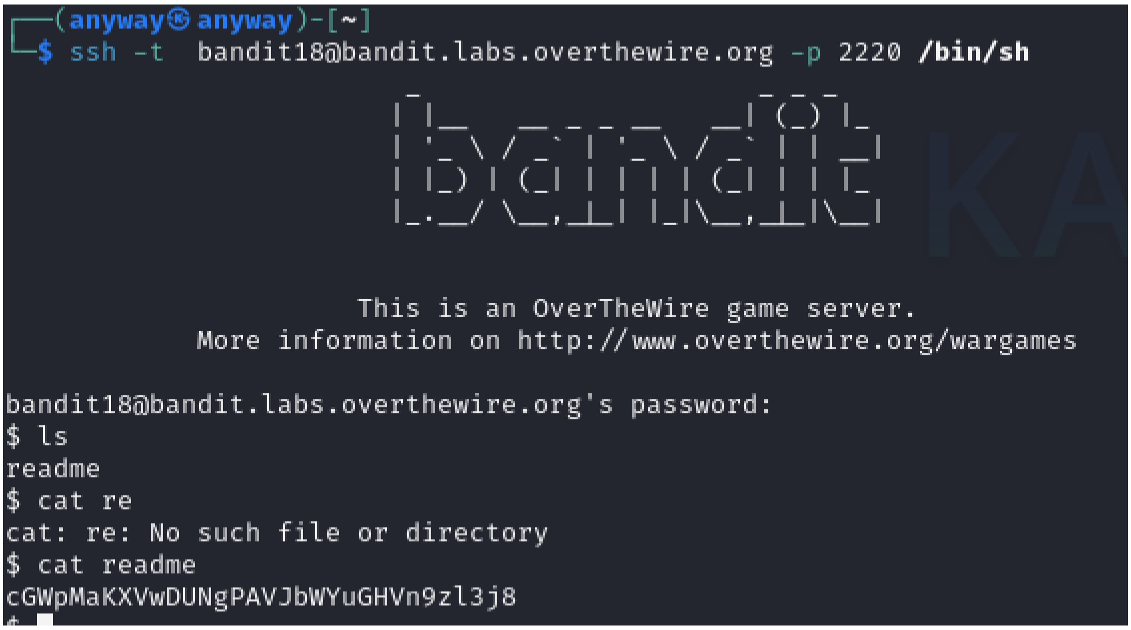
Level Goal

The password for the next level is stored in a file **readme** in the homedirectory. Unfortunately, someone has modified **.bashrc** to log you out when you log in with SSH.

Commands you may need to solve this level

ssh, ls, cat

So there’s a problem with bashrc, so we can not log in with ssh, instead, we’ll have to do differently:

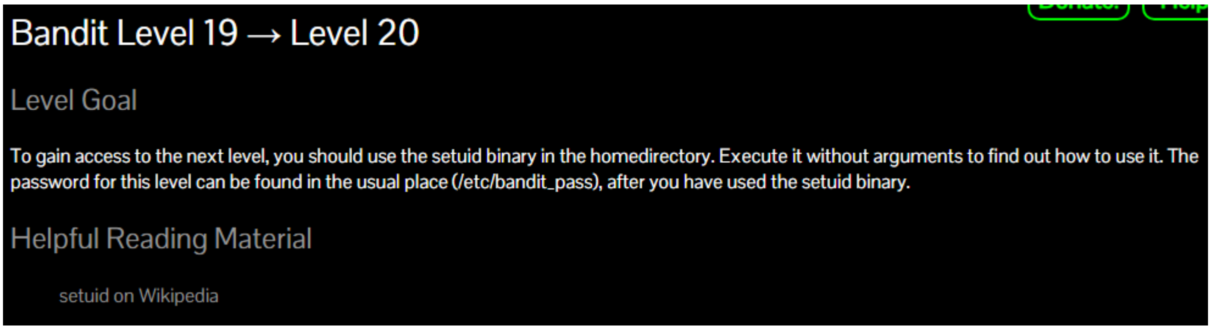


```
┌──(anyway㉿anyway)-[~]
└─$ ssh -t  bandit18@bandit.labs.overthewire.org -p 2220 /bin/sh
                         _                     _ _ _
                        | |__   __ _ _ __   __| (_) |_
                        | '_ \ / _` | '_ \ / _` | | __|
                        | |_) | (_| | | | | (_| | | |_
                        |_.__/ \__,_|_| |_|\__,_|_|\__|


                      This is an OverTheWire game server.
            More information on http://www.overthewire.org/wargames

bandit18@bandit.labs.overthewire.org's password:
$ ls
readme
$ cat re
cat: re: No such file or directory
$ cat readme
cGWpMaKXVwDUNgPAVJbWYuGHVn9zl3j8
$ 
```

We used SSH with -t, this option forces the SSH client to allocate a pseudo-terminal and not request a remote shell. And enter the same Bandit18 password. There will be a file, just cat that and we get the flag:

cGWpMaKXVwDUNgPAVJbWYuGHVn9zl3j8

Level 20:



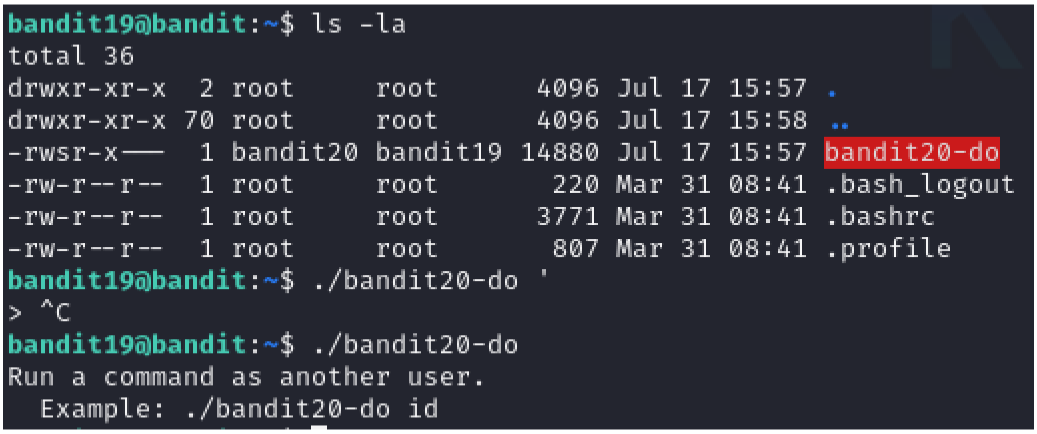
Bandit Level 19 → Level 20

Level Goal

To gain access to the next level, you should use the setuid binary in the homedirectory. Execute it without arguments to find out how to use it. The password for this level can be found in the usual place (/etc/bandit_pass), after you have used the setuid binary.

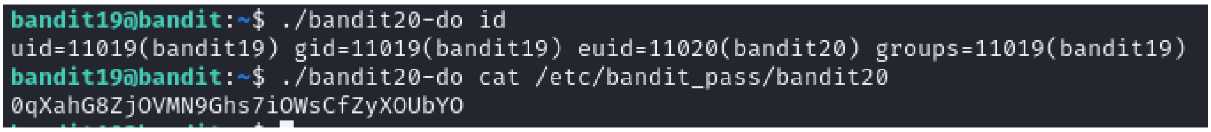
Helpful Reading Material

- setuid on Wikipedia



```
bandit19@bandit:~$ ls -la
total 36
drwxr-xr-x  2 root     root      4096 Jul 17 15:57 .
drwxr-xr-x 70 root     root      4096 Jul 17 15:58 ..
-rwsr-x---  1 bandit20 bandit19 14880 Jul 17 15:57 bandit20-do
-rw-r--r--  1 root     root       220 Mar 31 08:41 .bash_logout
-rw-r--r--  1 root     root      3771 Mar 31 08:41 .bashrc
-rw-r--r--  1 root     root       807 Mar 31 08:41 .profile
bandit19@bandit:~$ ./bandit20-do '
> ^C
bandit19@bandit:~$ ./bandit20-do
Run a command as another user.
  Example: ./bandit20-do id
```

We have a file which is in the use of another user.



```
bandit19@bandit:~$ ./bandit20-do id
uid=11019(bandit19) gid=11019(bandit19) euid=11020(bandit20) groups=11019(bandit19)
bandit19@bandit:~$ ./bandit20-do cat /etc/bandit_pass/bandit20
0qXahG8ZjOVMN9Ghs7iOWsCfZyXOUbYO
```

We just need to check the ID and then cat the bandit20 password as we know the location.

0qXahG8ZjOVMN9Ghs7iOWsCfZyXOUbYO